From Cognitive Binary Logic to Cognitive Intelligent Agents

N. Popescu-Bodorin*, IEEE Member , and V.E. Balas**, IEEE Senior Member
* Dept. of Mathematics and Computer Science, ‘Spiru Haret’ University, Bucharest, Romania
** Faculty of Engineering, ‘Aurel Vlaicu’ University, Arad, Romania
bodorin@ieee.org, balas@drbalas.ro

***Abstract*—The relation between *self awareness* and *intelligence* is an open problem these days. Despite the fact that *self awarness* is usually related to *Emotional Intelligence*, this is not the case here. The problem described in this paper is how to model an agent which *knows* (Cognitive) Binary Logic and which is also able to pass (without any mistake) a certain family of *Turing Tests* designed to verify its knowledge and its discourse about the modal states of truth corresponding to well-formed formulae within the language of Propositional Binary Logic.**

I. Introduction

The relation between *self awareness* and *intelligence* is an open problem these days. Despite the fact that *self awarness* is usually related to *Emotional Intelligence*, this is not the case here. The problem described in this paper is how to model an agent which *knows* (Cognitive) Binary Logic [1] and which is also able to pass (without any mistake) a certain family of *Turing Tests* [2] designed to verify its knowledge and its discourse about the modal states of truth [3] (necessary truth - denoted t, contextual truth - denoted c_t, impossibly truth / necessary false / contradiction - denoted f, [1]) corresponding to well-formed formulae within the language of Propositional Binary Logic (BPL).

The context of this paper is given by [4] and [1]. More precisely, in order to improve a complex software platform for iris recognition [4], an inference engine is needed. The computational model of this engine will be derived from Computational formalization of Cognitive Binary Logic ($CCBL$) introduced in [1]. First step in this direction is to extend $CCBL$ up to an intelligent agent enabled to pass some Turing tests, and this is the subject of the present paper.

II. Preparing for the Turing Test

In order to pass the Turing test, the agent must have conversational capacities. Let us assume that the agent gets the input p which *looks* like a well-formed formula of PBL. An example of this kind is the *Liar Paradox* discussed in [1]. The problem is that a deductive discourse [1] depends on the given input string but it also depends on the given goal which is obvious for a human agent, but not for a software agent.

In other words, for a software agent, the input string p can be translated into one of the following sentences: ‘p is (always) false’, ‘p is (always) true’, ‘p is a contextual truth’, ‘p is false and well-formed’, ‘p is true and well-formed’, or into one of the following queries: ‘is it p a theorem?’, ‘is it p a contradiction?’, ‘is it p a contextual truth?’.

A. The cognitive dialect

The introduction of two semantic markers denoted ‘(!):’ and ‘(?):’ is mandatory in order to differentiate between assertions (affirmations) and queries (questions), respectively. With these notations, in the cognitive dialect, the deductive discourses are derived from the deductive discourses written in $CCBL$ by adding ‘(!):’ or ‘(?):’ prefix to each vertices.

The second reason for introducing these markers is that in order to prove a certain degree of self awareness, an agent must be able to understand the difference between the assertions like ‘I ask’ and ‘I say’ and also between ‘I ask myself’, ‘I say to myself’ (‘I found’, ‘I proved’, ‘I know’), ‘I ask you/someone’, ‘I say to you/someone’.

When it comes to imagining a logical human-machine dialog, the most important thing is that if the human tells something to the agent, then what is told can be true or false, but anything said by the agent must be true (or else, it is inevitably that the agent is inconsistent and, sooner or later, it will fail to pass a certain Turing test).

Also, to keep the design of our agent as simple as possible, we will consider that all assertions are *positive*, i.e. all of them declare that something is true:
‘it is true that p’:

$$(!) : t \to (p \vee f), \tag{1}$$

or ‘it is true that p is false’,

$$(!) : t \to [(p \to f) \vee f], \tag{2}$$

or ‘it is true that p is a contextual truth’:

$$(!) : t \to [(c_t \to p) \vee f]. \tag{3}$$

By analogy, any query will ask for something true:
‘is it p true?’:

$$(?) : t \to (p \vee f), \tag{4}$$

or ‘is it true that p is false?’:

$$(?) : t \to [(p \to f) \vee f], \tag{5}$$

or ‘is it true that p is a contextual truth?’:

$$(?) : t \to [(c_t \to p) \vee f]. \tag{6}$$

The third convention allows the agent to manipulate all three states of modal truth using a purely binary vocabulary. We achieve this by introducing the *dialog function* (N. Popescu-Bodorin):

$$d : \widehat{CI} \to \{l, n\} \times \hat{t} \cup \hat{f}, \qquad (7)$$

where: l and n are two reserved labels (meaning that the input assertion/query is *logical* or *nonsense*, respectively), $\hat{t}$ is the class of all tautologies, $\hat{f}$ is the class of all contradictions, and $\widehat{CI}$ is defined as follows: if p is a well-formed formula of PBL ($p \in FORM$), then:

$$[(!) : t \to (p \vee f)] \in \widehat{CI},$$

$$[(!) : t \to ((p \to f) \vee f)] \in \widehat{CI},$$

$$[(?) : t \to (p \vee f)] \in \widehat{CI},$$

$$[(?) : t \to ((p \to f) \vee f)] \in \widehat{CI}.$$

The output of the dialog function d is computed using the following rules:

1) p is a tautology if and only if the full deductive discourse [1] of the input assertion

$$(!) : t \to (p \vee f)$$

is a deductive proof [1]. In this case, the dialog function outputs the doublet:

$$(l; t) \in \{l, n\} \times \hat{t}.$$

I.e. the agent proves that p is always true and the input assertion is (logically) well-formed.

2) p is a tautology if and only if the full deductive discourse of the input assertion

$$(!) : t \to ((p \to f) \vee f)$$

is a deconstruction [1]. In this case, the dialog function outputs the doublet:

$$(n; t) \in \{l, n\} \times \hat{t}.$$

I.e. the agent proves that p is always true and founds that asserting falsity for a tautology is a logical nonsense.

3) p is a contradiction if and only if the full deductive discourse of the input assertion

$$(!) : t \to (p \vee f)$$

is a deconstruction. In this case, the dialog function outputs the doublet:

$$(n; f) \in \{l, n\} \times \hat{f}.$$

I.e. the agent proves that p is always false and founds that asserting truth for a contradiction is a logical nonsense.

4) p is a contradiction if and only if the full deductive discourse of the input assertion

$$(!) : t \to ((p \to f) \vee f)$$

is a deductive proof. In this case, the dialog function outputs the doublet:

$$(l; f) \in \{l, n\} \times \hat{f}.$$

I.e. the agent proves that p is always false and the input assertion is well-formed.

5) p is a tautology if and only if the full deductive discourse of the input query

$$(?) : t \to (p \vee f)$$

is a deductive proof. In this case, the dialog function outputs the doublet:

$$(l; t) \in \{l, n\} \times \hat{t}.$$

I.e. the input question is well-formed, the agent proves that p is always true and gives a pozitive answer to the input query.

6) p is a tautology if and only if the full deductive discourse of the input assertion

$$(?) : t \to ((p \to f) \vee f)$$

is a deconstruction. In this case, the dialog function outputs the doublet:

$$(l; f) \in \{l, n\} \times \hat{f}.$$

I.e. the input question is well-formed, the agent proves that p is always true and gives a negative answer to the input query.

7) p is a contradiction if and only if the full deductive discourse of the input query

$$(?) : t \to (p \vee f)$$

is a deconstruction. In this case, the dialog function outputs the doublet:

$$(l; f) \in \{l, n\} \times \hat{f}.$$

I.e. the input question is well-formed, the agent proves that p is always false and gives a negative answer to the input query.

8) p is a contradiction if and only if the full deductive discourse of the input query

$$(?) : t \to ((p \to f) \vee f)$$

is a deductive proof. In this case, the dialog function outputs the doublet:

$$(l; t) \in \{l, n\} \times \hat{t}.$$

I.e. the input question is well-formed, the agent proves that p is always false and gives a pozitive answer to the input query.

9) p is a contextual truth if and only if none of the full deductive discourses of the input assertions

$$(!) : t \to (p \vee f)$$

$$(!) : t \to ((p \to f) \vee f)$$

or of the input queries

$$(?) : t \to (p \vee f)$$

$$(?) : t \to ((p \to f) \vee f)$$

is a deductive proof. In this case, the dialog function outputs the doublet:

$$(l; [p_t \to (t \to p)] \wedge [p_f \to (p \to f)])$$

I.e. the agent find a context p_t which makes the formula $(t \to p)$ satisfiable and also finds a context p_f which makes the formula $(p \to f)$ satisfiable.

By analyzing the outputs of the *dialog function* it can be seen that, in the *cognitive dialect* is legal to ask *anything* but it is illegal to assert falsity for a tautology or to assert truth for a contradiction.

B. Simple examples

Let us consider that the formula $\alpha = (p \to q)$ is given to be studied. If the input query is $(?) : t \to (\alpha \vee f)$, a full deductive discourse written in *cognitive dialect* would be:

$$(?) : t \to ((p \to q) \vee f)$$

$$(?) : (t \wedge p) \to (q \vee f)$$

The context which makes the formula $(t \to \alpha)$ satisfiable is:

$$\alpha_t = [(t \leftrightarrow q) \vee (p \leftrightarrow q) \vee (p \leftrightarrow f)].$$

If the input query is $(?) : t \to ((\alpha \to f) \vee f)$, a full deductive discourse written in *cognitive dialect* would be:

$$(?) : t \to (((p \to q) \to f) \vee f)$$

$$(?) : (t \wedge (p \to q)) \to f$$

$$(?) : [t \to (p \vee f)] \wedge (?) : [(t \wedge q) \to f]$$

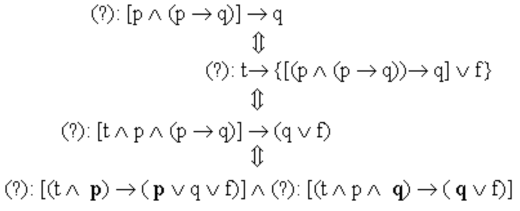
The context which makes the formula $(\alpha \to f)$ satisfiable is:

$$\alpha_f = [(t \leftrightarrow p) \wedge (q \leftrightarrow f)].$$

The output of the *dialog function* is the following doublet:

$$(l; [\alpha_t \to (t \to \alpha)] \wedge [\alpha_f \to (\alpha \to f)])$$

Hence, α is a contextual truth. Also, α_t and α_f describe the solutions of Boolean satisfiability problems $(t \to \alpha)$ and $(\alpha \to f)$, respectively.



$$(?): [p \wedge (p \to q)] \to q$$
$$\Updownarrow$$
$$(?): t \to \{[(p \wedge (p \to q)) \to q] \vee f\}$$
$$\Updownarrow$$
$$(?): [t \wedge p \wedge (p \to q)] \to (q \vee f)$$
$$\Updownarrow$$
$$(?): [(t \wedge \mathbf{p}) \to (\mathbf{p} \vee q \vee f)] \wedge (?): [(t \wedge p \wedge \mathbf{q}) \to (\mathbf{q} \vee f)]$$

Fig. 1. A deductive discourse for Modus Ponens written in the *cognitive dialect*.

In the second example, Modus Ponens is analyzed. For the input query:

$$(?) : [p \wedge (p \to q)] \to q,$$

a full deductive discourse written in *cognitive dialect* would be the deductive proof presented in Fig.1 and the output of the *dialog function* is the doublet (l, t).

In the third example, the contradiction $(p \wedge \neg p)$ is analyzed. For the input query:

$$(?) : t \to [((p \wedge \neg p) \to f) \vee f],$$

a full deductive discourse written in *cognitive dialect* would be the following deductive proof:

$$(?) : t \to [((p \wedge \neg p) \to f) \vee f],$$

$$(?) : (t \wedge p \wedge \neg p) \to f$$

$$(?) : (t \wedge p) \to (p \vee f)$$

The output of the *dialog function* is the doublet (l, t). Hence, the input assertion $(!) : t \to [(p \wedge \neg p) \vee f]$ will be recognized as a logical nonsense and the output of the *dialog function* will be the doublet (n, f).

The forth example analyze the Modus Tollens argument. For the input query:

$$(?) : (|p \to q| \wedge |\neg q|) \to |\neg p|$$

a full deductive discourse written in *cognitive dialect* would be the following deductive proof:

$$(?) : t \to \{[(|p \to q| \wedge |\neg q|) \to |\neg p|] \vee f\}$$

$$(?) : (t \wedge |p \to q| \wedge |\neg q|) \to (|\neg p| \vee f)$$

$$(?) : (t \wedge |p \to q| \wedge p) \to (q \vee f)$$

$$[(?) : (t \wedge p) \to (p \vee q \vee f)] \wedge [(?) : (t \wedge q \wedge p) \to (q \vee f)]$$

Hence, the output of the *dialog function* is the doublet (l, t).

In the fifth example, we consider the tautology:

$$[p \to (q \to r)] \to [(p \to q) \to (p \to r)].$$

A deductive discourse written in *cognitive dialect* is presented in Fig.2 and the output of the *dialog function* is the doublet (l, t).

III. The Agent

The basic functionality of the proposed Cognitive Intelligent Agent (CIA) is described in Fig.3. Let us consider the Turing tests containing the following type of problems: for an arbitrary formula $p \in FORM$, CIA is required to find if the input assertion/query is or isn't a logical nonsense, and also if p is a tautology, a contextual truth, or a contradiction.

Since the $CCBL$ theory is sound and complete [1], CIA will give the correct answer for any input query written in cognitive dialect. Also, if the input assertion is a logical

$$
\begin{gathered}
(?):[\,p \to (q \to r)\,] \to [\,(p \to q) \to (p \to r)\,] \\
\Updownarrow \\
(?):t \to \{\,[\,(\,p \to |q \to r|\,) \to (\,|p \to q| \to |p \to r|\,)\,] \vee f\,\} \\
\Updownarrow \\
(?):[\,t \wedge (\,p \to |q \to r|\,)\,] \to [\,(\,|p \to q| \to |p \to r|\,) \vee f\,] \\
\Updownarrow \\
(?):[\,t \wedge (\,p \to |q \to r|\,) \wedge |p \to q|\,] \to (\,|p \to r| \vee f\,) \\
\Updownarrow \\
(?):[\,t \wedge (\,p \to |q \to r|\,) \wedge |p \to q| \wedge p\,] \to (\,r \vee f\,) \\
\Updownarrow \\
\{(?):[\,t \wedge (p \to |q \to r|) \wedge \mathbf{p}] \to (\,\mathbf{p} \vee r \vee f)\,\} \wedge \{(?):[\,t \wedge (p \to |q \to r|) \wedge q \wedge p] \to (r \vee f)\} \\
\Updownarrow \\
\{(?):[t \wedge |q \to r| \wedge q \wedge p] \to (r \vee f\,)\} \wedge \{(?):(t \wedge q \wedge \mathbf{p}) \to (\,\mathbf{p} \vee r \vee f\,)\} \\
\Updownarrow \\
\{(?):[t \wedge \mathbf{q} \wedge p] \to (\,\mathbf{q} \vee r \vee f\,)\} \wedge \{(?):[t \wedge \mathbf{r} \wedge q \wedge p] \to (\,\mathbf{r} \vee f\,)\}
\end{gathered}
$$

Fig. 2. A deductive discourse written in *cognitive dialect* for the tautology $[p \to (q \to r)] \to[(p \to q) \to (p \to r)]$.

nonsense, CIA will correctly recognize it. Therefore, CIA will pass with a success rate of 100% all sessions of Turing tests designed to verify its knowledge and its discourse about the modal states of truth corresponding to formulae within the language of PBL. Hence, there is no doubt that as a software agent, CIA demonstrate the highest possible degree of intelligence. Still, the agent is not enabled to be aware of itself and of its environment or to simulate self-awareness.

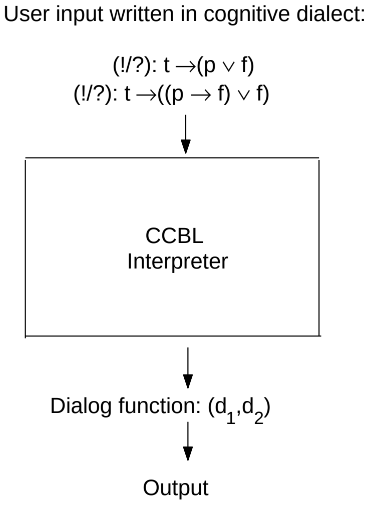


Fig. 3. The Cognitive Intelligent Agent

IV. Conclusion and Future Work

The basic design of an inteligent agent was proposed in this paper. It is an example of a fully intelligent agent which is not at all aware of itself. Still, it is enabled to engage in simple conversations about the modal states of truth of well-formed formulae of PBL, without doing any mistakes.

Future developments will include self-awarnes which is needed in order to enable the agent to supervize complex computations and to engage direct communication with humans on specific subjects, other than the modal truth state of the formulae of PBL. For example, we plan to gather humans‘ opinions about some particular iris recognition results in a similar manner with that described in [5] where customers‘ emotive responses to a product are collected using a questionnaire. The goal is corelate iris recognition results obtained automaticaly with human feedback and to explore the limitations that could apear in iris recognition when using eye images captured in insufficiently constrained aquisition conditions.

On the other hand, we know that some sub-problems of iris recognition are in NP (the class of problems solvable in nondeterministic polynomial time), and consequently heuristic algorithms must be used in order to achieve some speed. The problem is that quantifying the quality of their results is a *matter of degree* [6] (in fact, the results of these heuristic algorithms are *near solutions*, not exact solutions). Therefore, future developments in the direction of fuzzy logic are not excluded at all.

References

[1] N. Popescu-Bodorin, L. State, *Cognitive Binary Logic - The Natural Unified Formal Theory of Propositional Binary Logic*, accepted in The 4^{th} European Computing Conference (ECC 2010, Bucharest).

[2] A. M. Turing, *Computing machinery and intelligence*, Mind, 59, 433-460, 1950.

[3] P. Blackburn, M. de Rijke, Y. Venema, *Modal Logic* Cambridge University Press, 2000.

[4] N. Popescu-Bodorin, *Exploring New Directions in Iris Recognition*, Proc. 11^{th} International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC‘ 09,Conference Publishing Services - IEEE Computer Society, pp. 384-391, 2010.

[5] A. Mohais, A. Nikov, A. Sahai,S. Nesil, *A tunable swarm-optimization-based approach for affective product design*, Proc. 9th WSEAS Int. Conf. on Mathematical and Computational Methods in Science and Engineering, MACMESE'07, pp.254-258, 2007.

[6] L. A. Zadeh, *Test-Score Semantics for Natural Languages*, Proc. of the 9^{th} conference on Computational linguistics, Vol. 1, pp. 425 - 430, 1982.